

Design Specification

Chamo

By **Anonymous Messaging:**

Alex Kem

Angeline Dequit

Sopheak Chim

Darren Seng

Ivy Ly

Version: 3.0

May 2, 2024

Version Table:

Version 1	Date: 10/7/23	
Version 2	Date: 3/16/2024	
	<i>Section Changed</i>	<i>Page #</i>
	<i>Modify Feature #3</i>	<i>Page #8</i>
	<i>Modify UML Diagram</i>	<i>Page #10</i>
Version 3	Date: 5/2/2024	
	Summary of Changes: <ul style="list-style-type: none"> Added information is in green Updated information is in blue Deleted information is in grey and crossed out 	
	Section Changed	Page #
	Modify Executive Abstract	4
	Modify Feature #3: One-on-one Messaging	7
	Replace Feature #4: Notifications with Feature #4: Edit Profile, update description and activity diagram	8
	Update Site Map Diagram and summary	11

Table of Contents

Executive Abstract	4
Technologies Used	4
Reference	4
Behavior Specification	5
Feature #1: Picking a Topic/Interest for Matching.....	5
Figure 1: Picking a Topic/ Interest for Matching Activity Diagram.....	5
Feature #2: Private Profile Sharing.....	6
Figure 2: Private Profile Sharing Activity Diagram.....	6
Feature #3: One-on-One Message.....	7
Figure 3: One-on-One Message Activity Diagram	7
Feature #4: Notifications Profile Customization.....	8
Feature #5: Safety Filter	9
Figure 5: Safety Filter Activity Diagram	9
Architecture Specification	10
Class Diagram.....	10
Figure 6: UML Class Diagram	10
Site Map.....	11
Figure 7: Site Map (updated)	11

Executive Abstract

Written By: Ivy Ly

The document entails the behavior specifications and the architecture specifications for the web application, ~~Intrvt.io~~ **Chamo**. ~~Intrvt.io~~ **Chamo** offers a safe and welcoming platform allowing users to anonymously message one another. Ultimately, the web application aims to decrease users' judgment of other users at the beginning, and let users interact based on a ~~certain criterion (e.g., personality test or similar interest) or no criterion (randomization)~~ **a shared topic of interest (e.g Anime, Comics, Tabletop Games)**, and users will have the option to connect with another user on a "deeper level" ~~(sharing real profile)~~.

Behavior specifications show graphical representations of the application's most critical features that make it unique compared to other similar products. The activity diagrams display the sequence of activities from a start point to the end point, detailing the many decision paths that exist in the progression of events contained in a specific activity.

Architecture specifications visually present the class diagram and the site map of the web application. The class diagram is a blueprint of the system to model the classes that make up the system, to display the relationship between the classes, describe what those classes do, and the services provided. In other words, the diagram will describe the overall logic of classes that will be utilized in the development of the application. Additionally, the site map will display the overall flow of utilization of the web application by various users interacting with the unique features provided on the web application.

With the above ideas, it will help guide the design of ~~Intrvt.io~~ **Chamo**. The development team will be the main driving force behind this project to implement functional features that will provide users with optimal service and experience. Improvements will occur throughout the development process. Lastly, information used in this document was referenced from the Requirements Document to advance the development of the web application.

Technologies Used

- **Front-end:** React, Redux, React Router, TailwindCSS, and Axios
- **Back-end:** Node.js, Express.js, Websocket (Socket.io), Passport.js
- **Database management:** MongoDB and Mongoose
- **Password security and management:** Nodemailer and Bcrypt
- **Notifications, messaging, and data updates:** Firebase Cloud Messaging or Pusher Channels and Redis
- **API management:** JSON Web Tokens, GraphQL, Winston, Swagger

Reference

Requirement Specification Document: This document outlines the requirement specifications for Intrvt.io

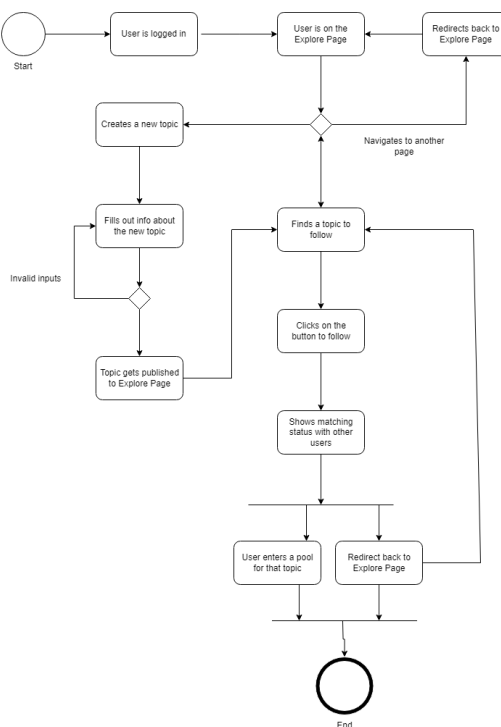
Behavior Specification

Feature #1: Picking a Topic/Interest for Matching

Written By: Alex Kem

When the user is signed up or logged in, the user is directed to the Explore Page. The Explore Page is essentially hosts where users can find other users with similar interests/topics. From there, we would match users together to start 1-on-1 chats with that user. The user can either stay on the explore page or navigate to other pages on our app. On the Explore Page, a user can browse the page to see what topics they are interested in. If the users find something that they like, they would be placed in a pool of users that are also interested in the topic. The system would select two people from that pool and pair them up for a 1-on-1 chat. However, if the system is unable to pair a user to someone – because the number of users in that pool is odd - it would wait for one more user to join that topic so that it can continue with the matching process. Additionally, users can create their own topics for other users to join. They would have to click on the ‘Create Topic Button’, fill out a few forms about the topic, and publish the topic to the Explore Page. The user would be redirected to the Explore Page.

Figure 1: Picking a Topic/ Interest for Matching Activity Diagram

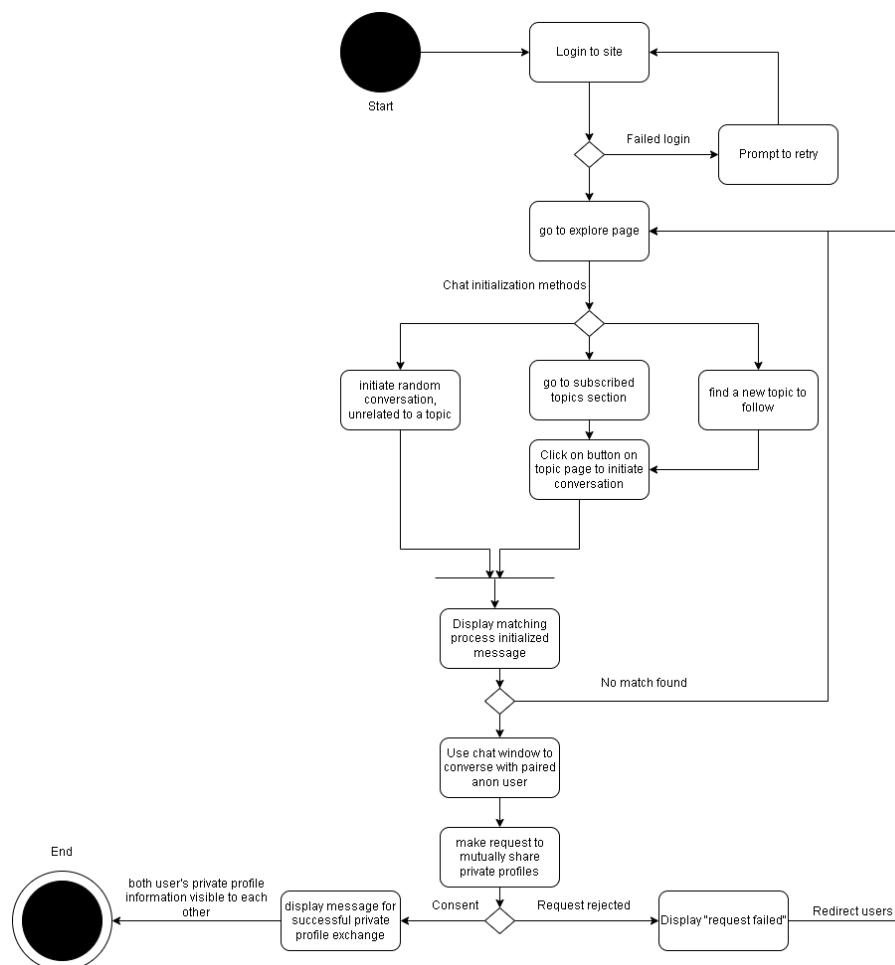


Feature #2: Private Profile Sharing

Written By: Darren Seng

After a user has initiated a conversation with another user, and both users mutually get to know each other through their chat, one user can make a request to mutually share their private profile information with the other person. At the time of the request, both users' profiles appear anonymous to each other. Both users also have private profiles that are only revealed to the other person if they have a mutual connection with them, so the private profile exchange request is meant to create that connection. For the request to go through, the other user has to accept. Upon accepting the request, both users can now see each other's private profiles. Private profiles have the required information for real name, age, profile picture, birthday, country, and have some optional information the profile's owner can enter at their discretion. The chat partner gets added to a contacts list. The chat partner's profile is no longer visible as "anonymous" to the current user.

Figure 2: Private Profile Sharing Activity Diagram

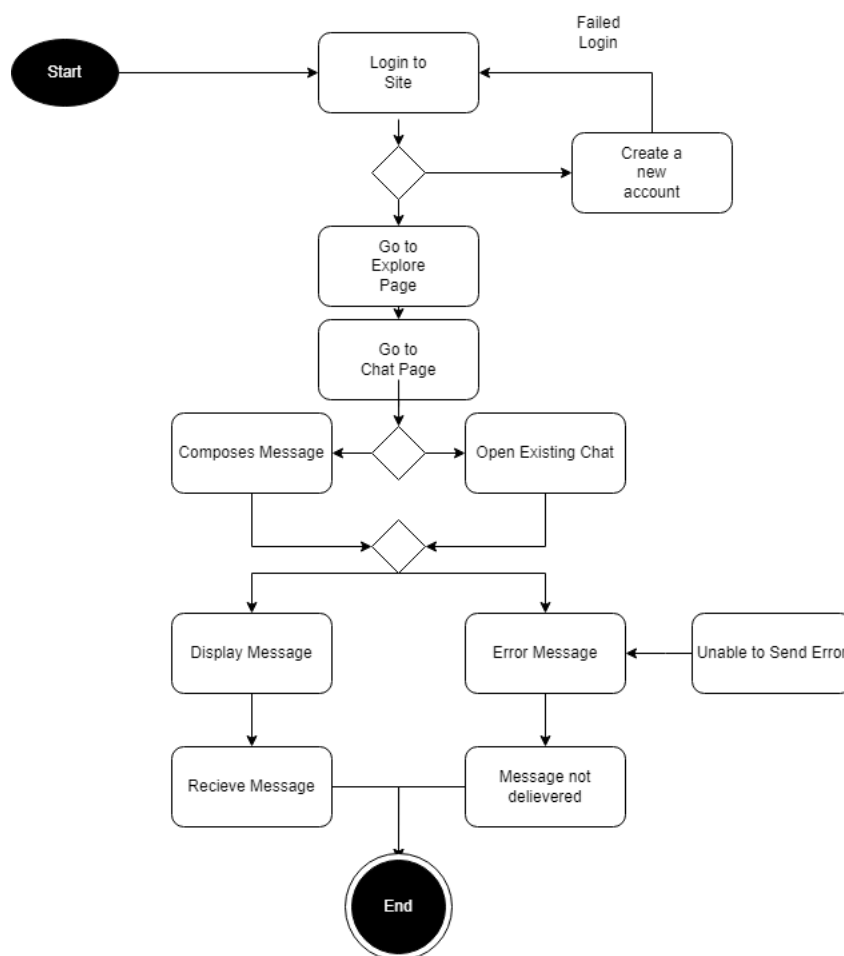


Feature #3: One-on-One Message

Written By: Sopheak Chim

Once matched and paired with another user, ~~using many initialization methods,~~ **a user is placed into a chatroom with the other user.** The user is then able to start messaging the user they are paired with. This interaction is like many other messaging applications where one can send **and** receive messages. Messages exchanged between both users will be in the form of a chat window where earlier messages are stored within the chat window. ~~Messages can be viewed, edited, deleted. The whole chat window can be removed, and~~ A user can have multiple active chats, which are saved in an archived list. For the one-on-one messaging, there will be a place for a user to compose their messages first. Composed messages must check if it can be sent or not. Once it can be sent, the message will be delivered to the other user and display it in their chat window. ~~If the message was not sent there will be an indication saying the message could not be delivered and it won't be displayed on the other user's chat window. It will then prompt a retry for the user to resend the message again.~~ Once the message is sent, received delivered, and displayed the one-on-one messaging feature is completed.

Figure 3: One-on-One Message Activity Diagram



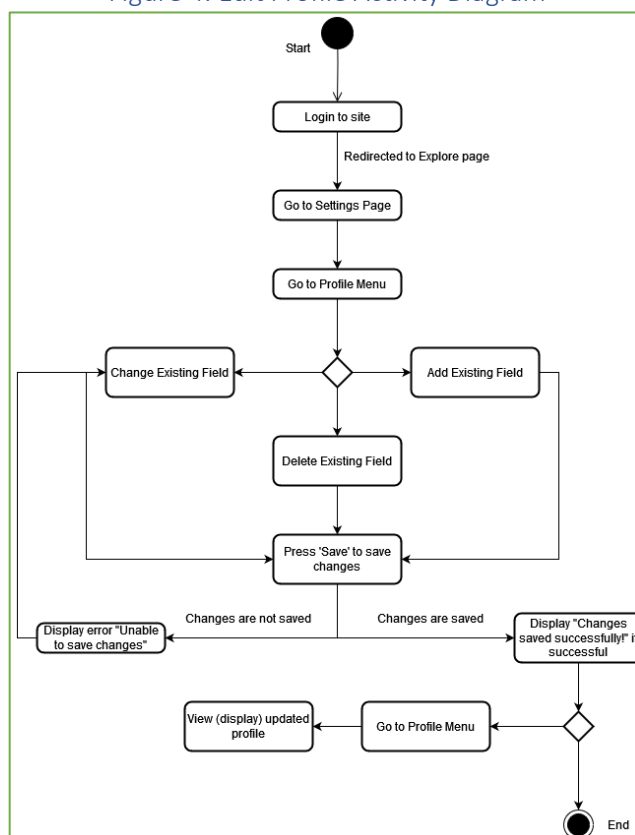
Feature #4: Profile Customization Notifications

Written By: Angeline Dequit

Our site will provide toggle-able notifications to the users. Users are able to toggle select notifications including but not limited to: new messages, new group messages, site updates, and submitted interest topics statuses. After clicking on the notification icon in the global navigation bar, the user will be able to view a list of their notifications. Users are able to view notifications about new chats, along with a link to a page to adjust their notification settings. Clicking on a link will bring the user to the page of the respective notification category. For example, if a user clicks on notification of type “New messages from User 2”, upon clicking the notification, User 1 will be brought to the chat window with User 2. In their notification settings display, users have the option of toggling all or certain notifications on or off.

Users will be able to customize their profile after they create an account. User are able to change their profile color icon, add a bio, their age, country, personality type (MBTI), and topics of interest. After users log in, users will navigate to the settings page. From there, they will click on the ‘Profile Menu’ option in the settings table of contents. Their profile settings can be adjusted by toggling the fields that they want to change. For example, after clicking on a country from the dropdown and pressing “Save”, their profile should be updated. Users can view their own profile by clicking on the ‘view profile’ button on the global navigation bar.

Figure 4: Edit Profile Activity Diagram

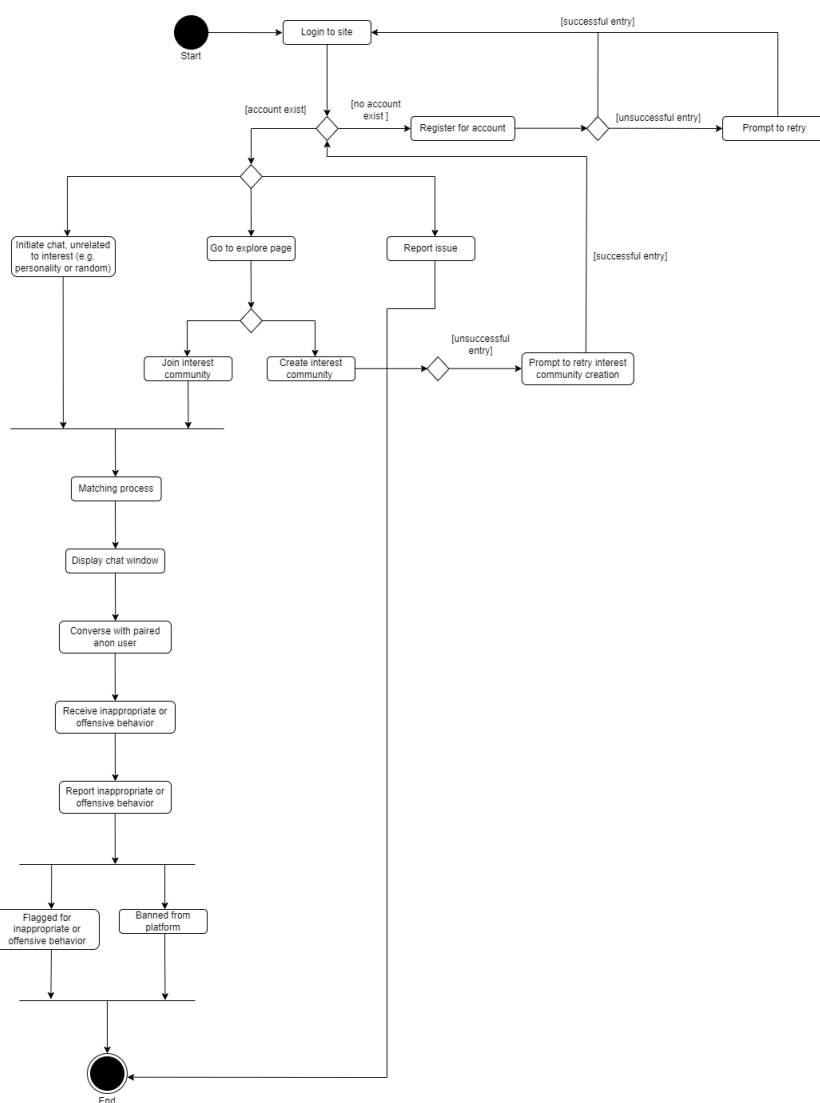


Feature #5: Safety Filter

Written By: Ivy Ly

The Safety Filter Activity Diagram shows how this feature is integrated within the system. The goal for this feature is to filter inappropriate or offensive language, allowing user a safe and friendly experience. First, the user will need to login or register an account. Upon account creation, the user will be prompted to reenter the entry field if the content is inappropriate or a duplication within the system. Same process for interest community creation. Then, the user will have access to the services provided on the platform. When conversing with other users, if a user is displaying inappropriate or offensive behavior, they will be reported for their actions, potentially leading to being flagged or banned from the platform. Additionally, the users will have the option to report other issues related to the application.

Figure 5: Safety Filter Activity Diagram



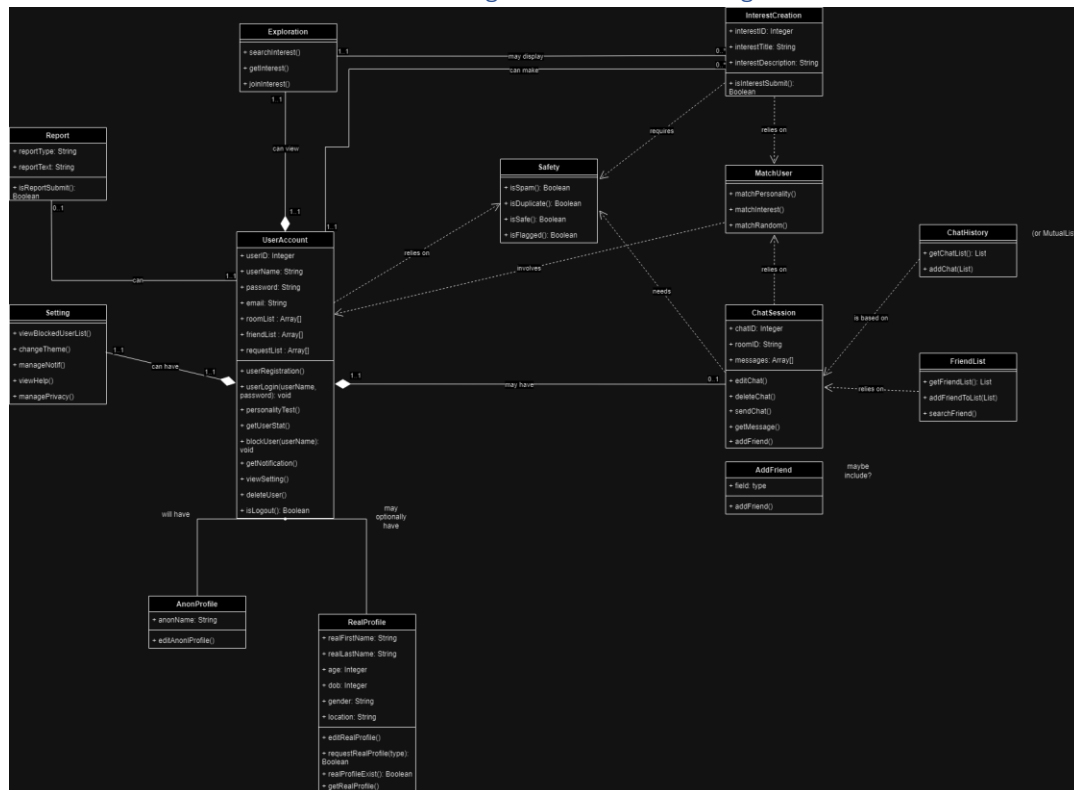
Architecture Specification

Class Diagram

Written and Contributed By: Sopheak Chim, Angeline Dequit, Alex Kem, Ivy Ly, and Darren Seng

Below is the tentative UML Class Diagram for **Chamo** web application. The diagram shows all of the classes needed to fully implement the program, and the relationship between these classes. First, the **UserAccount** class is the main, parent class that needs to be created to access the other classes. The **Exploration** class, **Chat** class, and **Setting** class have a composition relationship with the **UserAccount** class, meaning the **UserAccount** class must exist for the other classes to exist. The **UserAccount** class has an implementation relationship with the **AnonProfile** class and the **RealProfile** class. The **Exploration** class may show interest created by the user. The **Chat** class and **Exploration** class rely on the **MatchUser** class to match users together to start messaging. During the chatting process, users will have the option to request to reveal their real profile if the profile exists and there is mutual agreement from both parties. The **ChatHistory** class is based on the **Chat** class to have a list of chats the user is involved in. Users will have the option to add a friend after chatting with another user, which will display on their friends list. The user can report issues with the application. Most of the classes will rely on the **Safety** class.

Figure 6: UML Class Diagram



Site Map

Written and Contributed By: Angeline Dequit, Darren Seng

Below is the proposed site map for our **Chamo** website. Shown is the interface of our website, detailing its pages, links, and sections. Our diagram is color coded according to its respective category. Pages are the blue-colored chicklets. Links on a page are the green-colored chicklets. Sections of a page are the orange-colored chicklets. Refer to Figure 7 for the visualization of the site map:

Figure 7: Site Map (updated)

